

# A Multigrid Optimization Algorithm for a Class of Quasilinear Variational Inequalities of the Second Kind\*

Sergio González-Andrade and Sofía López

Research Center on Mathematical Modeling (MODEMAT) and

Departamento de Matemática - Escuela Politécnica Nacional

Ladrón de Guevara E11-253, Quito 170413, Ecuador

`sergio.gonzalez@epn.edu.ec` and `sofia.lopezo@epn.edu.ec`

March 10, 2017

## Abstract

In this paper we propose a multigrid optimization algorithm (MG/OPT) for the numerical solution of a class of quasilinear variational inequalities of the second kind. This approach follows from the fact that the solution of the variational inequality is given by the minimizer of a nonsmooth energy functional, involving the  $p$ -Laplace operator. Therefore, we propose a Huber regularization of the functional and a finite element discretization for the problem. Further, we analyze the regularity of the discretized energy functional, and we are able to prove that its Jacobian is slantly differentiable. This regularity property is useful to analyze the convergence of the MG/OPT algorithm. In fact, we demonstrate that the algorithm is globally convergent by using the mean value theorem for slantly differentiable functions. Finally, we analyze the performance of the MG/OPT algorithm when used to simulate the viscoplastic flow of Bingham, Casson and Herschel-Bulkley fluids in a pipe. Several numerical experiments are carried out to show the main features of the proposed method.

**Keywords:** Multigrid methods. Variational inequalities.  $p$ -Laplacian. Large-scale optimization. Preconditioned descent algorithms.

**AMS Subject Classification:** 65N55, 47J20, 36J62, 65K15, 65N30, 90C30, 76A05.

## 1 Introduction

In this paper, we focus on the development of a multigrid algorithm for the fast finite element solution of a class of quasilinear variational inequalities of the second kind. The main idea is the analysis of an efficient multigrid approach to this kind of problems which, usually, leads us to a family of large scale problems.

Let  $\Omega$  be an open and bounded set in  $\mathbb{R}^n$  with Lipschitz boundary  $\partial\Omega$ . We are concerned with the numerical solution of the following class of quasilinear variational inequalities of the second

---

\*Supported in part by the Escuela Politécnica Nacional del Ecuador, under the project PIMI 14-12 “Viscoplastic Fluids in Food Industry” and the MATH-AmSud Project “SOCDE-Sparse Optimal Control of Differential Equations: Algorithms and Applications”. This paper was developed within the Master Program in Optimization of the Mathematics Department at Escuela Politécnica Nacional del Ecuador.

kind: find  $u \in W_0^{1,p}(\Omega)$  such that

$$\int_{\Omega} |\nabla u|^{p-2} (\nabla u, \nabla(v-u)) dx + g \int_{\Omega} |\nabla v| dx - g \int_{\Omega} |\nabla u| dx \geq \int_{\Omega} f(v-u) dx, \forall v \in W_0^{1,p}(\Omega),$$

where  $1 < p < \infty$ ,  $g > 0$  and  $f \in L^q(\Omega)$ . Here,  $q = \frac{p}{p-1}$  stands for the conjugate exponent of  $p$ . It is known that the solution of these variational inequalities corresponds to a first order necessary optimality condition for the following class of nonsmooth optimization problems.

$$\min_{u \in W_0^{1,p}(\Omega)} J(u) := \frac{1}{p} \int_{\Omega} |\nabla u|^p dx + \int_{\Omega} |\nabla u| dx - \int_{\Omega} f u dx. \quad (1)$$

In consequence, we focus on the fast solution of this optimization problem.

The class of variational inequalities in study provide a versatile tool in the study of free boundary problems which arise in the modelling of complex physical phenomena. Diverse problems including the flow of electro- and magneto-rheological fluids, flow of viscoplastic materials and glaciology have been successfully simulated by this kind of models ([10, 24, 15]).

Several approaches have been proposed for the numerical solution of problems like (1). In [14] an Augmented Lagrangian method is implemented for the numerical simulation of the flow of viscoplastic materials. In [10], a preconditioned descent algorithm is proposed and analyzed both in finite and infinite dimension settings. Regarding the use of multigrid algorithms, in [16, 17] the author proposes algorithms for variational inequalities of the first and the second kind, based on extended relaxation methods. In [18], the author propose a multigrid algorithm for variational inequalities of the second kind using a combination of convex minimization with constrained Newton linearization. However, these contributions focus on variational inequalities involving linear elliptic operators such as the Laplacian.

The multigrid approach is a very appealing way to develop fast solution algorithms for the numerical approximation of (1). In fact, the numerical solution of this kind of problems leads us to a family of large scale problems, which involve an increasing number of variables. Since these problems are computationally expensive to solve, the multigrid algorithms provide an efficient way to manage all the data generated when discretizing the problem. Further, it is natural to look for an algorithm which, in the context of the multigrid approximation, focus on the direct optimization of the energy functional.

The multigrid optimization method (MG/OPT) corresponds to a nonlinear programming adaptation of the *full approximation storage* (FAS) scheme and was introduced, for instance in [20, 21], as an effective tool for large scale optimization problems. This algorithm works with different discretization levels of the optimization problem and take advantage of the coarse problems to generate search directions for the finer problems. Similar approaches have been used for problems involving quasilinear operators, such as the  $p$ -Laplace operator (see [2] and the references therein), but to the best of our knowledge, there are no contributions proposing a MG/OPT algorithm for variational inequalities of the second kind involving this kind of operators.

In this paper, we propose and analyze a MG/OPT algorithm to compute the finite element solution of a Huber regularized version of (1). Considering the structure of the optimization problem, specifically the low regularity of the functional, we use a class of preconditioned descent algorithms proposed in [10] as underlying optimization methods or smoothers. Further, the low regularity of the functional prevents us from doing a classic analysis of convergence. Therefore, we perform the convergence analysis of the MG/OPT algorithm by using a mean value theorem for slantly differentiable functions. Finally, we present a deep numerical experimentation focused on the numerical simulation of viscoplastic materials.

Let us mention that, although the method developed in this article is concerned with variational inequalities of the second term involving the nonsmooth term  $\int_{\Omega} |\nabla u| dx$ , the results can be extended to other variational inequalities of the second kind including more general convex functionals.

The paper is organized as follows: In section (2), we present several results on generalized differentiability, which will be used to analyze the convergence of the multigrid method. Since the problem is ill-posed, in section (3) we propose a local regularization for the objective functional. Further, we present the finite element discretization of the problem. The MG/OPT method is presented in section (4), whereas the convergence of the algorithm is discussed in section (5). In section (6), a brief explanation of the underlying optimization and line search algorithms is presented. In section (7), we analyze the behaviour of the proposed methodology when applied to the numerical simulation of viscoplastic flow. We perform several experiments in order to show the main features of the algorithm. Finally, in section (8), we outline conclusions on this work and discuss future contributions.

## 2 Preliminaries on Generalized Differentiability

In this section we introduce several results on generalized differentiability, such as the concept of slanting function, slant differentiability and the mean value theorem for slantly differentiable functions.

**Definition 2.1.** Let  $X$  and  $Y$  be two normed spaces,  $D$  be a nonempty open set in  $X$  and  $J : D \subset X \rightarrow Y$  be a given mapping. For  $x \in D$  and  $h \in X$ , if the limit

$$J'(x)(h) := \lim_{t \rightarrow 0} \frac{J(x + th) - J(x)}{t}$$

exists, the function is said to be directionally differentiable. Further,  $J'(x)(h)$  is the directional derivative of  $J$  at  $x$  in the direction  $h$ . If the directional derivative  $J'(x)(h)$  exists for all  $h \in X$  and  $J'(x)$  is a linear and continuous operator from  $X$  to  $Y$ , then  $J$  is said to be Gâteaux differentiable.

**Definition 2.2.** Let  $X$  and  $Y$  be Banach spaces, and  $D$  be an open domain in  $X$ . A function  $J : D \subset X \rightarrow Y$  is said to be slantly differentiable at  $x \in D$  if there exists a mapping  $J^\circ : D \rightarrow \mathcal{L}(X, Y)$  such that the family  $\{J^\circ(x + h)\}$  of bounded linear operators is uniformly bounded in the operator norm, for  $h$  sufficiently small, and

$$\lim_{h \rightarrow 0} \frac{J(x + h) - J(x) - J^\circ(x + h)h}{\|h\|} = 0.$$

The function  $J^\circ$  is called a slanting function for  $J$  at  $x$ .

**Definition 2.3.** Suppose that  $J^\circ : D \rightarrow L(X, Y)$  is a slanting function for  $J$  at  $x \in D$ . We call the set

$$\partial_S J(x) := \left\{ \lim_{x_k \rightarrow x} J^\circ(x_k) \right\}$$

the slant derivative of  $J$  associated with  $J^\circ$  at  $x \in D$ . Here,  $\lim_{x_k \rightarrow x} J^\circ(x_k)$  exists for any sequence  $\{x_k\} \subset D$  such that  $x_k \rightarrow x$ .

**Proposition 2.4.**

1. If  $J$  is continuously differentiable in  $D$ , we can set  $J^\circ(u) := J'(u) \forall u \in D$ . Thus,  $J'$  is a slanting function for  $J$  at every point of  $D$ .

2. A slantly differentiable function  $J$  at  $u$  can have infinitely many slanting functions at  $u$ . If  $J^\circ$  and  $H^\circ$  are both slanting functions for  $J$  at  $u \in D$ , then

$$P^\circ := \lambda J^\circ + (1 - \lambda)H^\circ$$

is also a slanting function for  $J$  at  $u \in D$ , where  $\lambda \in [0, 1]$ . Moreover,

$$\lim_{h \rightarrow 0} \|J^\circ(u + h)h - H^\circ(u + h)h\| = 0.$$

*Proof.* See [4, Sec.2] □

We now present an important example of a slanting differentiable function that will be useful in the subsequent sections.

**Example 2.5.** [11, Sec.3, Lemma 3.1] Let  $g > 0$  be a constant. The mapping

$$\vec{z} \rightarrow \max(g, \gamma|\vec{z}|)$$

from  $\mathbb{R}^n$  to  $\mathbb{R}$  is slantly differentiable on  $\mathbb{R}^n$ . Further, the slant derivative of this function is the characteristic function  $\chi_{A_\gamma}(\vec{z})$  defined by

$$\chi_{A_\gamma}(\vec{z}) = \begin{cases} 1, & \text{if } \vec{z} \in A_\gamma, \\ 0, & \text{if } \vec{z} \in X \setminus A_\gamma, \end{cases}$$

where  $A_\gamma := \{\vec{z} : \gamma|\vec{z}| \geq g\}$

Let us now introduce the mean value theorem for slantly differentiable functions.

**Theorem 2.6.** (Mean value theorem for slantly differentiable functions.) Let  $X$  and  $Y$  be Banach spaces, and  $D$  be an open domain in  $X$ . Suppose that  $J : D \subset X \rightarrow Y$  is slantly differentiable at  $x$ . Then for any  $h \neq 0$  such that  $x + h \in D$ , there exists a slanting function for  $J$  at  $x$  such that

$$J(x + h) - J(x) = J^\circ(x + h)h.$$

*Proof.* See [4, Sec. 2, Cor. 2.7, p. 1207]. □

### 3 Regularization and Discretization

The minimization problem (1) involves a convex non-smooth functional. The norm  $|\nabla y|$  in the second term leads us to a non-differentiable problem. We propose a local Huber regularization procedure in order to solve this issue. This regularization only changes locally the structure of the functional, preserving the qualitative properties. Huber regularization approach has been used in several contributions to analyze free boundary and nonsmooth problems (see [10] and the references therein).

Let  $\gamma > 0$ . We introduce the function  $\psi_\gamma : \mathbb{R}^n \rightarrow \mathbb{R}$  as follows:

$$\psi_\gamma : z \rightarrow \psi_\gamma(z) = \begin{cases} g|z| - \frac{g^2}{2\gamma} & \text{if } |z| > \frac{g}{\gamma} \\ \frac{\gamma}{2}|z|^2 & \text{if } |z| \leq \frac{g}{\gamma}. \end{cases}$$

The function  $\psi_\gamma$  corresponds to a local regularization of the Euclidean norm. Thanks to this procedure we obtain the following regularized optimization problem

$$\min_{u \in W_0^{1,p}(\Omega)} J_\gamma(u) := \frac{1}{p} \int_\Omega |\nabla u|^p dx + \int_\Omega \psi_\gamma(\nabla u) dx - \int_\Omega f u dx. \quad (2)$$

**Theorem 3.1.** *Let  $1 < p < \infty$  and  $\gamma > 0$ . Then, problem (2) has a unique solution  $u_\gamma \in W_0^{1,p}(\Omega)$ . Also, the sequence  $\{u_\gamma\} \subset W_0^{1,p}(\Omega)$  converges strongly in  $W_0^{1,p}(\Omega)$  to the solution  $\bar{u}$  of problem (1) as  $\gamma \rightarrow \infty$ .*

*Proof.* See [10, Sec. 2]. □

### 3.1 Finite element formulation

Let us introduce the finite element formulation of problem (2). Let  $\Omega_h$  be a triangulation of the domain  $\Omega$ ,  $n_e \in \mathbb{N}$  the number of triangles  $T_i$  such that  $\bar{\Omega}_h = \cup_{i=1}^{n_e} T_i$  and  $n$  the number of nodes of the triangulation  $\Omega_h$ . For any two triangles, their closures are either disjoint or have a common vertex or a common edge. Finally, let  $\{P_j\}_{j=1,\dots,n}$  be the vertices (nodes) associated with  $\Omega_h$ . Taking this into account, we define

$$V_h := \{v_h \in C(\bar{\Omega}_h) : v_h|_{T_i} \in \mathbb{P}_1, \forall T_i \in \Omega_h\},$$

where  $\mathbb{P}_1$  is the space of continuous piecewise linear functions defined on  $\Omega_h$ . Then the following space

$$V_h^0 = W_0^{1,p}(\Omega) \cap V_h \quad (3)$$

is the finite-dimensional space associated with the triangulation  $\Omega_h$ .

Considering the previous analysis, the finite element approximation of (2) is formulated as follows

$$\min_{u_h \in V_h^0} J_{\gamma,h}(u_h) := \frac{1}{p} \int_{\Omega_h} |\nabla u_h|^p dx + \int_{\Omega_h} \psi_\gamma(\nabla u_h) dx - \int_{\Omega_h} f u_h dx. \quad (4)$$

Hereafter, since we are working in finite dimensional spaces, the derivative of any discrete functional  $\mathcal{J}_h$  will be represented by the following notation  $\mathcal{J}'_h(u_h)(v_h) := \nabla \mathcal{J}_h(u_h)^\top v_h$ . Also, we adopt the notation  $\overset{\circ}{\nabla} \mathcal{J}_h(u_h)$  for a slanting function of  $\nabla \mathcal{J}_h(u_h)$ .

The convergence analysis of the multigrid algorithm is based on the differentiability properties of the functional  $J_{\gamma,h}(u)$ . In order to analyze the regularity of this discrete functional, we decompose it as follows

$$J_{\gamma,h}(u_h) := \mathcal{F}_h(u_h) + \mathcal{G}_{\gamma,h}(\nabla u_h), \quad (5)$$

where

$$\mathcal{F}_h(u_h) := \frac{1}{p} \int_{\Omega_h} |\nabla u_h|^p dx - \int_{\Omega_h} f u_h dx \text{ and } \mathcal{G}_{\gamma,h}(\nabla u_h) := \int_{\Omega_h} \psi_\gamma(\nabla u_h) dx.$$

$\mathcal{F}_h(u)$  is a functional associated to the discretized Dirichlet problem for the  $p$ -Laplace operator, and it is known to be a twice Gâteaux differentiable and strictly convex functional with ([1, 2, 8, 10])

$$\nabla \mathcal{F}_h(u_h)^\top v_h = \int_{\Omega_h} |\nabla u_h|^{p-2} (\nabla u_h, \nabla v_h) dx - \int_{\Omega_h} f v_h dx \quad \forall v \in V_h^0 \quad (6)$$

and

$$\begin{aligned}\nabla^2 \mathcal{F}_h(u_h)(v_h, w_h) &= \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla v_h \cdot \nabla w_h \, dx \\ &\quad + (p-2) \int_{\Omega_h} |\nabla u_h|^{p-4} (\nabla u_h \cdot \nabla v_h) (\nabla u_h \cdot \nabla w_h), \quad \forall v_h, w_h \in V_h^0.\end{aligned}\tag{7}$$

**Proposition 3.2.** *Let  $1 < p < \infty$ . The functional  $J_{\gamma,h}(u_h)$  is Gâteaux differentiable with*

$$\nabla J_{\gamma,h}(u_h)^\top v_h := \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla u_h \cdot \nabla v_h \, dx + g \int_{\Omega_h} \frac{\gamma(\nabla u_h \cdot \nabla v_h)}{\max(g, \gamma|\nabla u_h|)} \, dx - \int_{\Omega_h} f v_h \, dx \quad \forall v_h \in V_h^0. \tag{8}$$

Furthermore  $\nabla J_{\gamma,h}(u_h)$  is a slantly differentiable function with

$$\begin{aligned}\overset{\circ}{\nabla} J_{\gamma,h}(u_h)(v_h, w_h) &:= \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla v_h \cdot \nabla w_h \, dx \\ &\quad + (p-2) \int_{\Omega_h} |\nabla u_h|^{p-4} (\nabla u_h \cdot \nabla v_h) (\nabla u_h \cdot \nabla w_h) \, dx \\ &\quad + \int_{A_\gamma} g \frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|} \, dx - \int_{A_\gamma} g \frac{(\nabla u_h \cdot \nabla v_h) (\nabla u_h \cdot \nabla w_h)}{|\nabla u_h|^3} \, dx \\ &\quad + \int_{\Omega_{k-1} \setminus A_\gamma} \gamma (\nabla v_h \cdot \nabla w_h) \, dx, \quad \forall v_h, w_h \in V_h^0.\end{aligned}\tag{9}$$

*Proof.* Let us start by analyzing the functional  $\mathcal{G}_{\gamma,h}(\nabla u_h)$ . It is known that  $\mathcal{G}_{\gamma,h}$  is once Gâteaux differentiable (see [10, Sec. 2.2]), and moreover, we know that

$$\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)^\top v_h = g \int_{A_{\gamma,h}} \frac{\nabla u_h \cdot \nabla v_h}{|\nabla u_h|} \, dx + g \int_{\Omega_h \setminus A_{\gamma,h}} \gamma (\nabla u_h \cdot \nabla v_h) \, dx, \quad \forall v_h \in V_h^0,$$

where

$$A_{\gamma,h} = \{x \in \Omega_h : \gamma|\nabla u_h(x)| \geq g\}.$$

By using the max function, we can rewrite  $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)^\top v_h$  in the following way.

$$\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)^\top v_h := g \int_{\Omega_h} \frac{\gamma(\nabla u_h \cdot \nabla v_h)}{\max(g, \gamma|\nabla u_h|)} \, dx, \quad \forall v_h \in V_h^0. \tag{10}$$

Next, from (5), it follows that

$$\nabla J_{\gamma,h}(u_h)^\top v_h = \nabla \mathcal{F}_h(u_h)^\top v_h + \nabla \mathcal{G}_{\gamma,h}(\nabla u_h)^\top v_h, \tag{11}$$

which, thanks to (6) and (10), implies that

$$\nabla J_{\gamma,h}(u_h)^\top v_h := \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla u_h \cdot \nabla v_h \, dx + g \int_{\Omega_h} \frac{\gamma(\nabla u_h \cdot \nabla v_h)}{\max(g, \gamma|\nabla u_h|)} \, dx - \int_{\Omega_h} f v_h \, dx \quad \forall v_h \in V_h^0.$$

The last expression corresponds to the Gâteaux derivative of the discretized functional  $J_{\gamma,h}(u_h)$ . The second Gâteaux derivative of  $J_{\gamma,h}(u_h)$ , however, does not exist. In fact, the functional  $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)$  is not Gâteaux differentiable since this functional involves the max function. Fortunately, the max function is slantly differentiable when defined in finite dimensional spaces (see

Example 2.5). Thus, we can calculate the slant derivative of  $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)$ , denoted by  $\overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(\nabla u_h)$ , as follows.

$|\nabla u| \geq \frac{g}{\gamma}$ : Here, we have that

$$\begin{aligned} \overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(\nabla u_h)(v_h, w_h) &= g \int_{A_{\gamma,h}} \frac{\gamma(\nabla v_h \cdot \nabla w_h)}{\max(g, \gamma|\nabla u_h|)} dx - g \int_{A_{\gamma,h}} \frac{\chi_{A_{\gamma,h}}(x) \cdot \gamma(\nabla u_h \cdot \nabla w_h)}{(\max(g, \gamma|\nabla u_h|))^2 |\nabla u_h|} \gamma(\nabla u_h \cdot \nabla v_h) dx \\ &= g \int_{A_{\gamma,h}} \frac{\gamma(\nabla v_h \cdot \nabla w_h)}{\gamma|\nabla u_h|} dx - g \int_{A_{\gamma,h}} \frac{\gamma^2(\nabla u_h \cdot \nabla w_h)(\nabla u_h \cdot \nabla v_h)}{(\gamma|\nabla u_h|)^2 |\nabla u_h|} dx \\ &= g \int_{A_{\gamma,h}} \frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|} dx - g \int_{A_{\gamma,h}} \frac{(\nabla u_h \cdot \nabla w_h)(\nabla u_h \cdot \nabla v_h)}{|\nabla u_h|^3} dx, \end{aligned}$$

where  $\chi_{A_{\gamma,h}}$  is the slant derivative of function  $\max(g, \gamma|\nabla u_h|)$ .

$|\nabla u| < \frac{g}{\gamma}$ : Here, we have that

$$\begin{aligned} \overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(\nabla u_h)(v_h, w_h) &= g \int_{I_{\gamma,h}} \frac{\gamma(\nabla v_h \cdot \nabla w_h)}{\max(g, \gamma|\nabla u_h|)} dx - g \int_{I_{\gamma,h}} \frac{\chi_{A_{\gamma,h}}(x) \cdot \gamma(\nabla u_h \cdot \nabla w_h)}{(\max(g, \gamma|\nabla u_h|))^2 |\nabla u_h|} \gamma(\nabla u_h \cdot \nabla v_h) dx \\ &= g \int_{I_{\gamma,h}} \frac{\gamma(\nabla v_h \cdot \nabla w_h)}{g} dx \\ &= \int_{I_{\gamma,h}} \gamma(\nabla v_h \cdot \nabla w_h) dx, \end{aligned}$$

where  $I_{\gamma,h} := \Omega_h \setminus A_{\gamma,h}$ .

Then, the slant derivative of  $\nabla \mathcal{G}_{\gamma,h}(\nabla u_h)$  reads as follows

$$\begin{aligned} \overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(u_h)(v_h, w_h) &= g \int_{A_{\gamma,h}} \frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|} - g \int_{A_{\gamma,h}} \frac{(\nabla u_h \cdot \nabla v_h)(\nabla u_h \cdot \nabla w_h)}{|\nabla u_h|^3} \\ &\quad + \int_{I_{\gamma,h}} \gamma(\nabla v_h \cdot \nabla w_h), \quad \forall v_h, w_h \in V_h^0. \end{aligned} \tag{12}$$

On the other hand, from (7) we have that

$$\begin{aligned} \nabla^2 \mathcal{F}_h(u_h)(v_h, w_h) &= \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla v_h \cdot \nabla w_h dx \\ &\quad + (p-2) \int_{\Omega_h} |\nabla u_h|^{p-4} (\nabla u_h \cdot \nabla v_h)(\nabla u_h \cdot \nabla w_h), \quad \forall v_h, w_h \in V_h^0. \end{aligned} \tag{13}$$

Hence, from (11) and Proposition (2.4), we can state that

$$\overset{\circ}{\nabla} J(u_h)(v_h, w_h) = \nabla^2 \mathcal{F}_h(u_h)(v_h, w_h) + \overset{\circ}{\nabla} \mathcal{G}_{\gamma,h}(\nabla u_h)(v_h, w_h) \quad \forall v_h, w_h \in V_h^0, \tag{14}$$

which, thanks to (12), (13) and (14), yields that

$$\begin{aligned} \overset{\circ}{\nabla} J(u_h)(v_h, w_h) &= \int_{\Omega_h} |\nabla u_h|^{p-2} \nabla v_h \cdot \nabla w_h dx \\ &\quad + (p-2) \int_{\Omega_h} |\nabla u_h|^{p-4} (\nabla u_h \cdot \nabla v_h)(\nabla u_h \cdot \nabla w_h) dx \\ &\quad + g \int_{A_{\gamma,h}} \frac{(\nabla v_h \cdot \nabla w_h)}{|\nabla u_h|} dx - g \int_{A_{\gamma,h}} \frac{\nabla u_h \cdot \nabla v_h (\nabla u_h \cdot \nabla w_h)}{|\nabla u_h|^3} dx \\ &\quad + \gamma \int_{I_{\gamma,h}} (\nabla v_h \cdot \nabla w_h) dx, \quad \forall v_h, w_h \in V_h^0. \end{aligned} \tag{15}$$

□

## 4 The Multigrid for Optimization (MG/OPT) Method

In this section, we present the multigrid for optimization (MG/OPT) algorithm for solving the regularized and discretized optimization problem (4).

The MG/OPT method was introduced as an efficient tool for large scale optimization problems (see [21, 20]). The idea of the algorithm is to take advantage of the solutions of problems discretized in coarse meshes to optimize problems in finer meshes. The efficient resolution of coarse problems provide a way to calculate search directions for problems discretized in finer levels.

In order to present the algorithm, we shall introduce the following preliminaries. Let  $\{\Omega_k\}_{k=0,\dots,m}$  be a sequence of partitions of  $\Omega_h$ . As usual,  $\Omega_k$  is obtained from  $\Omega_{k-1}$  by a regular subdivision: the procedure joins the edge midpoints of any triangle in mesh  $\Omega_{k-1}$  by edges, and forms the new triangles of  $\Omega_k$  (see Figure 1). This procedure guarantees that the discretization parameters  $h_k$  and  $h_{k-1}$  satisfy that  $h_k = \frac{1}{2}h_{k-1}$ , which implies that  $\Omega_{k-1} \subset \Omega_k$ . Furthermore, we have that  $V_{k-1}^0 \subset V_k^0$ , for  $k = 1, \dots, m$ , where  $V_k^0$  is the FEM space associated to  $\Omega_k$  (see, for instance, [2, 9, 19]).

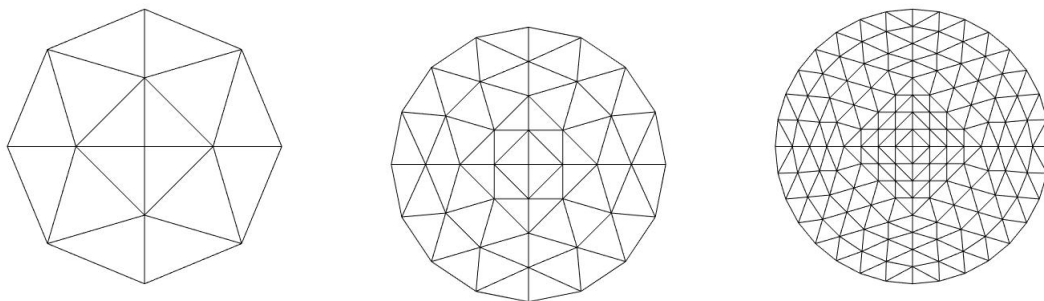


Figure 1: Regular subdivision of meshes (from left to right):  $\Omega_0$ ,  $\Omega_1$  and  $\Omega_2$ .

The multigrid approach involves several auxiliary operators that need to be introduced. As we are working with a set of meshes and the algorithm runs at each level of discretization, we need to transfer information among the different grids.

Then, we introduce the *fine-to-coarse grid transfer operator*,  $I_k^{k-1}$ , and the *coarse-to-fine grid transfer operator*,  $I_{k-1}^k$ . As the name implies, the *coarse-to-fine grid transfer operator* transfers information from the coarse mesh  $\Omega_k$  to the finer mesh  $\Omega_{k-1}$ . It is also called the prolongation operator. On the other hand, the *fine-to-coarse grid transfer operator* or restriction operator transfers information from the fine grid to the coarse one.

In this work we use the mesh data structure and the transfer operators presented in [9, Ch. 6, Ch. 13] to implement the operators  $I_k^{k-1}$  and  $I_{k-1}^k$ . This data structure describes the linear Lagrange triangles of a mesh and allows us to recover the index of a given node and identify its node parents in any mesh created by regular subdivision.

Taking this into account, we are ready to introduce the transfer operators. For the *coarse-to-fine grid transfer operator*,  $I_{k-1}^k$ , let denote  $\vec{v}_{k-1}$  as the vector containing the nodal values of a piecewise linear function on  $\Omega_{k-1}$ . The idea of this operator is to obtain the vector  $\vec{v}_k$  that contains the nodal values of the piecewise linear function on  $\Omega_k$ . Let us illustrate the procedure in a single triangle.



Suppose that we have the nodal values of its three nodes. If we refine it we obtain four triangles and six nodes as is shown in Figure 2. In order to obtain the nodal values of the midpoints with indexes 4, 5 and 6, we compute the average of the extreme nodes values. For instance, the value of the node with index 4 is the average of the nodal values with indexes 1 and 2. This procedure is repeated for all the midpoints on the finer mesh  $\Omega_k$ .



Figure 2: Refinement of a single triangle

Due to the fact that each node in  $\Omega_k$  either belongs to  $\Omega_{k-1}$  or is the midpoint of an edge in  $\Omega_{k-1}$ , the transfer algorithm computes the vector of nodal values for inner and boundary nodes denoted by  $\check{v}_k$  in  $\Omega_k$ . At the end, the algorithm extracts the components corresponding to the boundary nodes to obtain  $\vec{v}_k$ .

Next, we present the algorithm to compute  $\vec{v}_k$ . Let  $N_k$  be the vector that contains the indices of nodes in  $\Omega_k$  that are not in  $\Omega_{k-1}$ .

---

**Algorithm 1** Coarse-to-fine grid transfer operator  $I_{k-1}^k$

---

- 1: Copy the components of  $\vec{v}_{k-1}$  that corresponds to the components of  $\check{v}_k$ .
- 2: For  $i \in N_k$  do

$$(\check{v}_k)_i = \frac{1}{2} \left( (\check{v}_k)_{i_{end_1}} + (\check{v}_k)_{i_{end_2}} \right).$$

- 3: Extract the components of  $\check{v}_k$  corresponding to the boundary nodes and finally obtain  $\vec{v}_k$ , (the inner nodes of  $\check{v}_k$ ).
- 

Here  $i_{end_1}$  and  $i_{end_2}$  denote the indices of the endpoints of the edge in  $\Omega_{k-1}$  of which  $(\check{v}_k)_i$  is the midpoint.

For the *fine-to-coarse grid transfer operator*,  $I_k^{k-1}$ , let us recall that the vector  $\vec{v}_k$  contains the nodal values of a piecewise linear function on  $\Omega_k$ . Then, the vector with the nodal values of a piecewise linear function associated to the indices in  $N_k$  will be denoted as  $\check{v}_k$ . In this case, we perform the inverse procedure than the stated before. We present the following algorithm in order to illustrate it.

---

**Algorithm 2** Fine-to-coarse grid transfer operator  $I_k^{k-1}$ 


---

- 1: Copy the components of  $\vec{v}_k$  that corresponds to the components of  $\check{v}_k$ .
- 2: Initialize to zero the other components of  $\check{v}_k$ .
- 3: For  $i \in N_k$  do

$$(\check{v}_k)_{i_{end_1}} = \frac{1}{2} (\check{v}_k)_i \text{ and } (\check{v}_k)_{i_{end_2}} = \frac{1}{2} (\check{v}_k)_i,$$

- 4: Extract the components of  $\check{v}_k$  corresponding to the nodes in  $\Omega_{k-1}$ , except the boundary nodes, to obtain  $\vec{v}_{k-1}$ .
- 

The mesh data structure implemented allows us to copy and extract nodes from any specific discretization. We refer the reader to see [9, Ch. 13, sec. 13.2.1] for a detailed explanation of the transfer operators.

In multigrid schemes is standard to assume that

$$I_k^{k-1} = c \left( I_{k-1}^k \right)^T,$$

where  $c$  is a constant. In our case, the restriction and prolongation operators satisfy the condition (see [9, Sec. 13.2.1, p. 294])

$$I_k^{k-1} = \left( I_{k-1}^k \right)^T. \quad (16)$$

Once we have introduced the interpolation operators, we are ready to discuss the MG/OPT algorithm for problem (4). The MG/OPT method corresponds to a nonlinear programming adaptation of the *full approximation storage* (FAS) scheme (see [3, 25]). The multigrid subproblems arising from the different discretization levels are nonlinear optimization problems [20]. Then, MG/OPT is related to different optimization techniques ranging from the gradient method to quasi Newton methods to solve the problems at each level. The multigrid for optimization approach makes minimal requests about the underlying optimization algorithm. However, it is important to highlight that at each level of discretization we need to find a well estimated solution for the minimization subproblem. Then, the election of the underlying optimization algorithm is not trivial and depends on the inner characteristics of the optimization problem. As our goal is to solve problem (4), we use a class of descent algorithms as the underlying algorithm. This choice was made based on the structure of problem (4). As the  $p$  - Laplacian is involved in the functional, we have to consider that its finite element approximation derives in a highly nonlinear and degenerate finite dimensional problem [12]. Also, the functional  $J_{\gamma,h}$  involves a semismooth regular function. Then, the class of descent algorithms chosen is suitable to deal with this issue.

As we mentioned before, the main idea of the MG/OPT algorithm is to use coarse problems to generate, recursively, search directions for finer problems. Then, a line search procedure, along with the underlying optimization algorithm is used to improve the solution at each level of discretization.

In what follows we present the MG/OPT algorithm. The underlying optimization algorithm will be denoted by  $S_{opt}$  inside the multigrid approach. The initial discretized problem is given on the finest grid. To facilitate the implementation of the algorithm, the MG/OPT scheme is presented in a recursive formulation. Hence, we introduce the following slightly different notation for the optimization problem

$$\min_{u_k} \left( \hat{J}_{\gamma,k}(u_k) - f_k^\top u_k \right).$$

We set  $f_k = 0$  at the finest level  $k = m$ .  $\hat{J}_{\gamma,k}$  corresponds to the functional  $J_\gamma$  discretized at each level  $k = 1, \dots, m$ . Hereafter,  $u_k$  stands for  $u_{h_k}$ .

Summarizing, the algorithm reads as follows.

---

**Algorithm 3** MG/OPT recursive( $\nu_1, \nu_2$ ).

---

**if**  $k = 0$  **then**, solve  $\min_{u_k} \left( \hat{J}_{\gamma,k}(u_k) - f_k^\top u_k \right)$  and return.

**end if**

Otherwise,  $k > 0$ .

Pre-optimization: Apply  $\nu_1$  iterations of the optimization algorithm to the problem at level  $k$ .

$$u_k^\ell = S_{opt}(u_k^{\ell-1}), \quad \ell = 1, \dots, \nu_1.$$

Coarse-grid correction.

- Restrict:  $u_{k-1}^{\nu_1} = I_k^{k-1} u_k^{\nu_1}$ .
- Compute the fine-to-coarse gradient correction:

$$\tau_{k-1} := \nabla \hat{J}_{\gamma,k-1}(u_{k-1}^{\nu_1}) - I_k^{k-1} \nabla \hat{J}_{\gamma,k}(u_k^{\nu_1}).$$

- Define  $f_{k-1} := I_k^{k-1} f_k + \tau_{k-1}$   
and apply one cycle of MG/OPT( $\nu_1, \nu_2$ ) to

$$\min_{u_{k-1}} \left( \hat{J}_{\gamma,k-1}(u_{k-1}) - f_{k-1}^\top u_{k-1} \right)$$

to obtain  $\tilde{u}_{k-1}$ .

Coarse-to-fine minimization.

- Prolongate error:  $e := I_{k-1}^k (\tilde{u}_{k-1} - u_{k-1}^{\nu_1})$ .
- Line search in  $e$  direction to obtain a step size  $\alpha_k$ .
- Calculate the coarse-to-fine minimization step:  $u_k^{\nu_1+1} = u_k^{\nu_1} + \alpha_k e$ .

Post-optimization: Apply  $\nu_2$  iterations of the optimization algorithm to the problem at level  $k$ .

$$u_k^\ell = S_{opt}(u_k^{\ell-1}), \quad \ell = \nu_1 + 2, \dots, \nu_1 + \nu_2 + 1.$$


---

The algorithm presented above contemplates one iteration of a V-cycle initialized with a rough estimate of the solution on the finest grid.

## 5 Convergence Analysis

In this section, we discuss the convergence properties of Algorithm 3. Following [20, 21], we can state that the global convergence of the underlying optimization algorithm ensures global convergence of the MG/OPT method. This comes from the fact that if we have an approximated solution (given by the underlying optimization algorithm) at each discretization level, the algorithm generates search directions for problems discretized in finer meshes. Once we have the descent direction, a line search procedure is used to improve the solution at each finer problem.

In the classic convergence analysis of the MG/OPT methods [20, 21, 26], the three following conditions are critical.

1. The optimization problem keeps the convexity property at each level of discretization.

2. The subproblems

$$\min_{u_{k-1}} \left( \hat{J}_{\gamma,k-1}(u_{k-1}) - f_{k-1}^\top u_{k-1} \right) \quad (17)$$

are solved accurately enough.

3. The transfer operators satisfy the standard condition  $I_k^{k-1} = c(I_{k-1}^k)^T$ .

These last conditions are helpful to prove that the search direction provided by the MG/OPT algorithm is indeed a descent direction, *i.e.*,  $e$  satisfies that  $\nabla J_{\gamma,k}(u_k)^\top e < 0$ ,  $\forall k = 0, \dots, m$ . Further, the convexity condition is key to prove that the Hessian is positive definite. However, in our case, the usual Hessian does not exist. We will use the slantly differentiability of the functional  $J_{\gamma,h}$  to perform our convergence analysis.

Next, we state some comments about the previous conditions in our problem. The coarse grids subproblems correspond to the discrete optimization problem over the nested spaces  $V_{k-1}^0 \subset V_k^0$ . Hence, the inclusion preserves the convexity of the subproblems.

Since we perform a few iterations of a suitable globally convergent optimization algorithm ( $S_{opt}$ ), we ensure that the subproblems  $\min_{u_{k-1}} \left( \hat{J}_{\gamma,k-1}(u_{k-1}) - f_{k-1}^\top u_{k-1} \right)$  are solved accurately enough. Finally, (16) yields that  $I_k^{k-1} = (I_{k-1}^k)^T$ .

Let us recall that the descent direction for the MG/OPT algorithm is denoted by  $e$  and search directions of the underlying optimization algorithm, (inside the MG/OPT loop) are denoted by  $w_k$ .

Once we have discussed the convergence conditions, we introduce the following theorem of convergence for the MG/OPT Algorithm 3.

**Theorem 5.1.** *Suppose that the following hypothesis are satisfied:*

- The optimization algorithm,  $S_{opt}$ , applied to an optimization problem of any resolution, is globally convergent, *i.e.*,

$$\lim_{k \rightarrow \infty} \|\nabla J_{\gamma,h}(u_k)\| = 0. \quad (18)$$

- At least one of the parameters  $\nu_1$  or  $\nu_2$  is positive.
- The search direction  $e = I_{k-1}^k(\tilde{u}_{k-1} - u_{k-1}^{\nu_1})$  is a descent direction.

Then, MG/OPT algorithm is globally convergent in the same sense than (18).

*Proof.* The global convergence of the underlying optimization algorithm ensures the global convergence of MG/OPT since the approximated solution given at each level improves at every cycle of the multigrid optimization algorithm. Hence, we need to guarantee that the direction  $e$ , provided by the MG/OPT algorithm, is a descent direction. *i.e.*, we have to prove that

$$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e < 0, \quad \forall k = 0, \dots, m. \quad (19)$$

From this point, for the sake of readiness of the proof, we drop the subscript  $\gamma$ . First note that, if we solve

$$\min_{u_{k-1}} \left( J_{k-1}(u_{k-1}) - \tau_{k-1}^\top u_{k-1} \right)$$

exactly, then

$$\nabla J_{k-1}(\tilde{u}_{k-1}) - \tau_{k-1} = 0.$$

Since we are solving the problem approximately (or accurately enough), then we have that

$$\nabla J_{k-1}(\tilde{u}_{k-1}) - \tau_{k-1} = z, \quad (20)$$

for some  $z$  as small as the algorithm accuracy allows us. From Algorithm 3 we have that

$$\tau_{k-1} := \nabla J_{k-1}(u_{k-1}^{\nu_1}) - I_k^{k-1} \nabla J_k(u_k^{\nu_1}).$$

Hence, we can rewrite (20) as follows

$$\nabla J_{k-1}(\tilde{u}_{k-1}) = \nabla J_{k-1}(u_{k-1}^{\nu_1}) - I_k^{k-1} \nabla J_k(u_k^{\nu_1}) + z. \quad (21)$$

Thus, from (21) we have that

$$\begin{aligned} \nabla J_k(u_k^{\nu_1})^\top e &= \nabla J_k(u_k^{\nu_1})^\top I_{k-1}^k (\tilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\ &= \nabla J_k(u_k^{\nu_1})^\top (I_k^{k-1})^\top (\tilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\ &= (I_k^{k-1} \nabla J_k(u_k^{\nu_1}))^\top (\tilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\ &= (\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\tilde{u}_{k-1}) + z)^\top (w_{k-1}) \\ &= (\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\tilde{u}_{k-1}))^\top (w_{k-1}) + z^\top w_{k-1}, \end{aligned} \quad (22)$$

where

$$w_{k-1} = \tilde{u}_{k-1} - u_{k-1}^{\nu_1}.$$

Next, let us focus on the two first terms in the right-hand side of (22).

$$(\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\tilde{u}_{k-1}))^\top w_{k-1} = \nabla J_{k-1}(u_{k-1}^{\nu_1})^\top w_{k-1} - \nabla J_{k-1}(\tilde{u}_{k-1})^\top w_{k-1}. \quad (23)$$

We know, from Proposition 3.2, that  $\nabla J_{k-1}$  is slantly differentiable. Thus, Theorem 2.6 and Proposition 2.4 allow us to state that

$$\begin{aligned} -(\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\tilde{u}_{k-1}))^\top w_{k-1} &= (\nabla J_{k-1}(\tilde{u}_{k-1}) - \nabla J_{k-1}(u_{k-1}^{\nu_1}))^\top w_{k-1} \\ &= \overset{\circ}{\nabla} J_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1}), \end{aligned} \quad (24)$$

where  $\overset{\circ}{\nabla} J_{k-1}(\tilde{u}_{k-1})$  is given by (9). Furthermore, following the decomposition presented in (14), we know that

$$\overset{\circ}{\nabla} J_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1}) = \nabla^2 \mathcal{F}_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1}) + \overset{\circ}{\nabla} \mathcal{G}_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1}), \quad (25)$$

It is well known that  $\mathcal{F}$  is a strictly convex functional, which implies that ([1, 2, 8, 10])

$$\nabla^2 \mathcal{F}_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1}) \geq 0, \quad \forall w_{k-1} \in V_{k-1}^0 \setminus \{0\}. \quad (26)$$

Next, let us recall the expression  $\overset{\circ}{\nabla} \mathcal{G}_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1})$  given by

$$\begin{aligned} \overset{\circ}{\nabla} \mathcal{G}_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1}) &= \int_{A_{\gamma, h}} g \frac{(\nabla w_{k-1} \cdot \nabla w_{k-1})}{|\nabla \tilde{u}_{k-1}|} - \int_{A_{\gamma, h}} g \frac{(\nabla \tilde{u}_{k-1} \cdot \nabla w_{k-1})^2}{|\nabla \tilde{u}_{k-1}|^3} \\ &\quad + \int_{\Omega_{k-1} \setminus A_{\gamma, h}} \gamma (\nabla w_{k-1} \cdot \nabla w_{k-1}), \quad \forall w_{k-1} \in V_{k-1}^0. \end{aligned} \quad (27)$$

Applying Cauchy-Schwarz to the right hand side in (27), we have that

$$\begin{aligned} \int_{A_{\gamma}} g \frac{(\nabla \tilde{u}_{k-1} \cdot \nabla w_{k-1})^2}{|\nabla \tilde{u}_{k-1}|^3} &\leq \int_{A_{\gamma}} g \frac{|\nabla \tilde{u}_{k-1}|^2 |\nabla w_{k-1}|^2}{|\nabla \tilde{u}_{k-1}|^3} \\ &= \int_{A_{\gamma}} g \frac{|\nabla w_{k-1}|^2}{|\nabla \tilde{u}_{k-1}|} \\ &= \int_{A_{\gamma}} g \frac{(\nabla w_{k-1} \cdot \nabla w_{k-1})}{|\nabla \tilde{u}_{k-1}|}, \end{aligned}$$

which implies that

$$(\overset{\circ}{\nabla} \mathcal{G}_{k-1}(\tilde{u}_{k-1})(w_{k-1}, w_{k-1})) \geq \int_{\Omega_{k-1} \setminus A_\gamma} \gamma(\nabla w_{k-1} \cdot \nabla w_{k-1}) > 0, \text{ since } w_{k-1} \neq 0. \quad (28)$$

Summarizing (24), (25), (26) and (28) yield that

$$(\nabla J_{k-1}(u_{k-1}^{\nu_1}) - \nabla J_{k-1}(\tilde{u}_{k-1}))w_{k-1} < 0. \quad (29)$$

To check that  $e$  is a descent direction, we still need to prove that the third term of the right hand side in (22) satisfies that

$$z^\top w_{k-1} = z^\top (\tilde{u}_{k-1} - u_{k-1}^{\nu_1}) < 0.$$

Note that  $\tilde{u}_{k-1}$  is the solution of the problem

$$\min_{u_{k-1}} \left( J_{k-1}(u_{k-1}) - \tau_{k-1}^\top u_{k-1} \right).$$

Therefore,

$$J_{k-1}(\tilde{u}_{k-1}) - \tau_{k-1}^\top \tilde{u}_{k-1} < J_{k-1}(u_{k-1}^{\nu_1}) - \tau_{k-1}^\top u_{k-1}^{\nu_1},$$

which is equivalent to

$$J_{k-1}(\tilde{u}_{k-1}) - J_{k-1}(u_{k-1}^{\nu_1}) < \tau_{k-1}^\top (\tilde{u}_{k-1} - u_{k-1}^{\nu_1}), \quad (30)$$

since the optimization algorithm was initialized with  $u_{k-1}^{\nu_1}$ . Further, the functional  $J_{k-1}$  is also slantly differentiable since it is Gâteaux differentiable and, moreover  $J_k^\circ = \nabla J_k$  (see Proposition 2.4). Thus, we can take its Gâteaux derivative as a slanting function of  $J_{k-1}$ . Then again, from Theorem 2.6, we have that

$$(J_{k-1}(\tilde{u}_{k-1}) - J_{k-1}(u_{k-1}^{\nu_1})) = \nabla J_{k-1}(\tilde{u}_{k-1})^\top (w_{k-1}). \quad (31)$$

Hence, from the inequality (30) and equation (31) we have that

$$\begin{aligned} \nabla J_{k-1}(\tilde{u}_{k-1})(w_{k-1}) &< \tau_{k-1}^\top (\tilde{u}_{k-1} - u_{k-1}^{\nu_1}) \\ &= \tau_{k-1}^\top w_{k-1} \end{aligned}$$

which implies that

$$\nabla J_{k-1}(\tilde{u}_{k-1})w_{k-1} - \tau_{k-1}^\top w_{k-1} < 0. \quad (32)$$

Next, from (20) and (32) we obtain that

$$z^\top w_{k-1} < 0. \quad (33)$$

Then, from (22), (29) and (33) we have that

$$\nabla J_k(u_k^{\nu_1})^\top e < 0,$$

and we can conclude that  $e$  is a descent direction.

Finally, if at least one of the parameters  $\nu_1$  or  $\nu_2$  is positive, at least one iteration of the optimization algorithm is performed at every cycle of the MG/OPT. Consequently, as the underlying optimization algorithm is globally convergent, the multigrid optimization algorithm is globally convergent.  $\square$

## 6 Implementation

### 6.1 Optimization algorithm

In this section, we briefly discuss the underlying optimization algorithms. We implement the MG/OPT algorithm using two versions of the steepest descent algorithm: the gradient method [22] and the preconditioned descent algorithm proposed in [10].

Generally speaking, a descent method starts with an initial point  $u_0$  and, with information of first order, the algorithm finds directions that lead us to the minimum of the objective functional. Also, the method must find the length of the step,  $\alpha_r$ , along the chosen direction,  $w_r$ . The basic idea consists in finding  $\alpha_r$  and  $w_r$  such that:

$$J(u_r + \alpha_r w_r) < J(u_r), \text{ for } \alpha_r > 0$$

in each iteration of the method.

In the gradient method, the search direction  $w_r$  is determined by

$$w_r = -\nabla J(u_r).$$

On the other hand, in [10] the author takes into account that the behaviour of  $J(u_r)$  depends on the value of  $p$ . Then, the difficulties associated to the structure of the  $p$ -Laplacian are considered. Therefore, the search direction  $w_r$ , for the preconditioned descent algorithm, is determined by solving the following equation

$$P_r(w_r, v) = -\nabla J(u_r)^\top v, \quad \forall v \in V_h^0,$$

where the form  $P_r : V_h^0 \times V_h^0 \rightarrow \mathbb{R}$  is chosen as a variational approximation of the  $p$ -Laplacian.

In what follows we present the general preconditioned descent algorithm.

---

**Algorithm 4** General Preconditioned descent algorithm

---

- 1: Initialize  $u_0 \in V_h^0$  and set  $r = 0$ . If  $\nabla J(u_r) = 0$ , STOP. Otherwise:
- 2: Find a descent direction  $w_r$  by solving the following equation

$$P_r(w_r, v) = -\nabla J(u_r)^\top v, \quad \forall v \in V_h^0,$$

if  $1 < p < 2$ ,

$$P_r(w_r, v) = \int_{\Omega_h} (\epsilon + |\nabla u_r|)^{p-2} \nabla w_r \nabla v \, dx, \quad \forall v \in V_h^0,$$

else if  $p > 2$ ,

$$P_r(w_r, v) = \int_{\Omega_h} \nabla w_r \nabla v \, dx, \quad \forall v \in V_h^0,$$

end.

- 3: Perform an efficient line search technique to obtain  $\alpha_r$ .
  - 4: Update  $u_{r+1} := u_r + \alpha_r w_r$  and set  $r = r + 1$ .
- 

In [10] it is proved that the Algorithm 4 is globally convergent, both in finite and infinite dimension settings. For a deeper analysis of this algorithm, we refer the reader to [10, Sec. 3].

## 6.2 Line search technique

In this section, we describe the line search algorithm which will be used in the implementation of Algorithm 3. This algorithm uses polynomial models of the objective functional for backtracking, and it was originally proposed in [6, Sec. 6.3.2].

The algorithm reads as follows

---

**Algorithm 5** Line search algorithm by polynomial models

---

Let  $\sigma_1 \in (0, \frac{1}{2})$  and set  $\alpha_0 = 1$ .

- 1: Decide wheter  $J(u_k + \alpha_k) > J(u_k) + \sigma_1 \alpha_k \nabla J(u_k)^\top w_k$  holds. If so, STOP and set  $\alpha_k = \alpha_0$ . If not:
  - 2: Decide wheter steplength is too small. If so, STOP and terminate algorithm: routine failed to locate satisfactory  $x_{k+1}$  sufficiently distinct from  $x_k$ . If not:
  - 3: Decrease  $\alpha$  by a factor between 0.1 and 0.5 as follows:
  - 4: On the first backtrack: set  $\alpha_k := \tilde{\alpha}_2 = \operatorname{argmin} m_2(\alpha)$ , but constrain the new  $\alpha_k$  to be  $\geq 0.1$ .
  - 5: On all the subsequent backtracks: set  $\alpha_k := \tilde{\alpha}_3 = \operatorname{argmin} m_3(\alpha)$ , but constraint the new  $\alpha_k$  to be in  $[0.1\alpha_p, 0.5\alpha_p]$ .
  - 6: Return to step 1.
- 

If we set

$$\varphi_k(\alpha) := J(u_k + \alpha w_k),$$

then the quadratic model  $m_2$  is given by

$$m_2(\alpha) := (\varphi_r(1) - \varphi_r(0) - \varphi'_r(0))\alpha^2 + \varphi'_r(0)\alpha + \varphi_r(0),$$

while the cubic model  $m_3$  is given by

$$m_3(\alpha) := c\alpha^3 + d\alpha^2 + \varphi'_r(0)\alpha + \varphi_r(0),$$

where

$$\begin{pmatrix} c \\ d \end{pmatrix} = \frac{1}{\alpha_p - \alpha_{2p}} \begin{pmatrix} \frac{1}{\alpha_p^2} & \frac{-1}{\alpha_{2p}^2} \\ \frac{-\alpha_{2p}}{\alpha_p^2} & \frac{\alpha_p}{\alpha_{2p}^2} \end{pmatrix} \begin{pmatrix} \varphi_r(\alpha_p) - \alpha_r(0) - \alpha'_r(0)\alpha_p \\ \alpha_r(\alpha_{2p}) - \alpha_r(0) - \alpha'_r(0)\alpha_{2p} \end{pmatrix}$$

and  $\alpha_p$  and  $\alpha_{2p}$  are the last two previous values of  $\alpha_k$ . For further details and examples see [6, pp. 126-129] and the references therein, and [10].

## 7 Applications to the Numerical Simulation of Viscoplastic Flow

In this section we present the application of the MG/OPT method to numerical simulation of the steady flow of viscoplastic fluids. These materials are characterized by the existence of a yield stress [5, 7, 10, 14, 13]. This implies that the viscoplastic material exhibits no deformation if the shear stress imposed does not exceed the yield stress, i.e., it behaves as an ideal rigid solid. On the other hand, if the shear stress overpasses the yield stress, the material will deform as a nonlinear viscous fluid in most of the cases. In fact, Herschel - Bulkley and Casson fluids present a nonlinear stress-shear rate relationship, while Bingham fluids behave as a viscous fluid with linear stress-shear rate relationship (see Figure 3). Summarizing, the existence of the yield stress makes the flow of these materials to present rigid zones, known as the plug flow, and yielded zones.



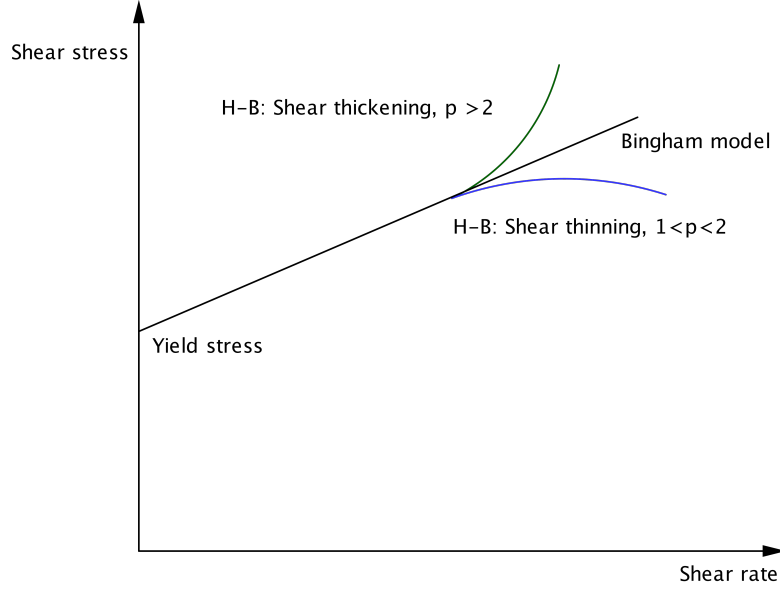


Figure 3: Viscoplastic models

In this work we simulate the pipe flow for the classic viscoplastic models: Herschel-Bulkley, Bingham and Casson. The three models fit the kind of nonsmooth optimization problems under study. In fact, it is well known that the velocity field of the flow across the cross-section of a pipe can be approximated by the solution of the following discretized minimization problem:

$$\min_{u_h \in V_h^0} J_h(u_h) := \phi(\nabla u_h) + \int_{\Omega_h} \psi_\gamma(\nabla u_h) dx - \int_{\Omega_h} f u_h dx, \quad (34)$$

where

$$\phi(\nabla u_h) = \begin{cases} \frac{1}{p} \int_{\Omega_h} |\nabla u_h|^p dx, & \text{for Herschel-Bulkley model} \\ \frac{1}{2} \int_{\Omega_h} |\nabla u_h|^2 dx, & \text{for Bingham model} \\ \frac{1}{2} \int_{\Omega_h} |\nabla u_h|^2 dx + \frac{4}{3} \sqrt{g} \int_{\Omega_h} |\nabla u_h|^{\frac{3}{2}} dx, & \text{for Casson model.} \end{cases}$$

In the coming numerical experiments, we present the results of the MG/OPT algorithm applied to the numerical solution of (34). For the computations in the MG/OPT algorithm we implement a V-cycle scheme with  $\nu_1 = \nu_2 = 2$  as the pre and post optimization iterations. In all algorithms the stopping criteria is fixed at a tolerance of  $10^{-7}$ . We perform the numerical experiments in two types of domains: unit square and unit circle. Also, we compare the performance of the MG/OPT algorithm with the performance of the underlying optimization algorithm when solving the same problem in the finest grid.

In the following tables we summarize the information of the different grids used at each level  $k$  of the V-cycle implemented for the MG/OPT algorithm.

Mesh ( $\Omega_k$ )	Nodes ( $n_k$ )	Elements ( $n_e$ )
$\Omega_6$	8321	16384
$\Omega_5$	2113	4096
$\Omega_4$	545	1024
$\Omega_3$	145	256
$\Omega_2$	41	64
$\Omega_1$	13	16

Table 1: Unit circle mesh information.

Mesh ( $\Omega_k$ )	Nodes ( $n_k$ )	Elements ( $n_e$ )
$\Omega_5$	4225	8192
$\Omega_4$	1089	2048
$\Omega_3$	289	512
$\Omega_2$	81	128
$\Omega_1$	25	32

Table 2: Unit square mesh information.

### 7.1 Herschel-Bulkley: case $1 < p < 2$

Herschel-Bulkley fluids are power-law materials with plasticity. The behaviour of these fluids depends on the value of  $p$ , which plays the role of the flow index. If  $1 < p < 2$  the material exhibits a pseudoplastic or shear-thinning behaviour. On the other hand, if  $p > 2$  the behaviour is shear-thickening (see Figure 3). Thanks to this index, the power-law model has been widely used to characterize several materials that include liquid foams, whipped cream, fluid foods, silly putty and some polymers [5].

In the following experiments, we compute the velocity field for a Herschel-Bulkley material for  $1 < p < 2$  in a pipe, considering circle and square cross sections. In these experiments we established the preconditioned descent algorithm (see [10]) as the underlying optimization algorithm  $S_{opt}$ . We set  $\epsilon = 10^{-6}$ .

#### Experiment 1

In this experiment we set the following parameters,  $p = 1.75$  and  $f = 1$ . In Table 3, each row represents one experiment. For each experiment, we present the finest and coarsest mesh, the number of V-cycles of the MG/OPT algorithm until the stopping tolerance is achieved, the tolerance reached and the execution time of the algorithm i.e., the CPU time when the stopping criteria is achieved.

g	Finest mesh	Coarsest mesh	V-cycles	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
0.2	$\Omega_6$	$\Omega_1$	11	7.38e-07	965.49	0.0527
	$\Omega_5$	$\Omega_1$	15	3.75e-07	258.9	0.0520
	$\Omega_4$	$\Omega_1$	13	4.70e-07	51.2	0.0523
	$\Omega_3$	$\Omega_1$	10	3.75e-07	10.04	0.0522
0.4	$\Omega_6$	$\Omega_1$	26	1.08e-07	3914.61	0.0043
	$\Omega_5$	$\Omega_1$	17	3.79e-07	329.07	0.0046
	$\Omega_4$	$\Omega_1$	13	7.23e-07	51.33	0.0047
	$\Omega_3$	$\Omega_1$	23	5.00 e-07	24.07	0.0037

Table 3: Results of the resolution of problem (34) with  $p = 1.75$ ,  $\gamma = 10^3$  and  $f = 1$ .

This experiment was initialized with the solution of the Poisson problem,  $-\Delta u_h = f$ . From Table 3 we can notice that the number of V-cycles is similar when solving the problem at the different levels of discretization. In Figure 5 we can see the decay of the norm  $|\nabla J_\gamma^\top e|$ . It behaves typically as in a steepest descent algorithm. However, it decays faster in the last iterations. This behaviour is inherited from the preconditioned descent algorithm (see [10, Sec. 4.3.1]). The resulting velocity

field is displayed in Figure 4. The flattening of the velocity in the center of the pipe corresponds to the plug flow velocity, where the material presents rigid zones.

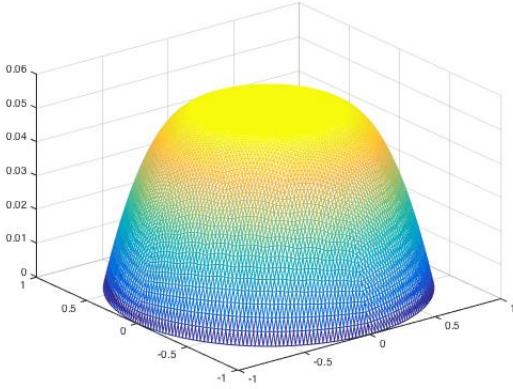


Figure 4: Calculated velocity  $u$  for mesh  $\Omega_6$ . Parameters:  $p = 1.75, g = 0.2, \gamma = 10^3$  and  $\epsilon = 10^{-6}$

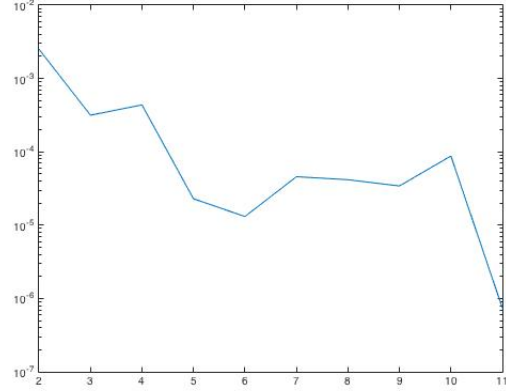


Figure 5: Calculated  $|\nabla J_\gamma^\top e|$  for mesh  $\Omega_6$  and  $g = 0.2$ .

In Tables 4, 5, 6 and 7, we present the performance of the line search globalization technique, described in Section 6.2, in 4 V-cycles randomly chosen. We consider  $g = 0.2$ ,  $\Omega_6$  as the finest mesh and  $\Omega_1$  as coarsest one. The experiment finished after 11 V-cycles, and we perform 5 line search procedures at each cycle. Also, the second column of each table let us observe that  $e$  is, indeed, a descent direction at each level.

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	$\alpha_k$	l.s it
$\Omega_1 - \Omega_2$	-1.36e-11	1	0
$\Omega_2 - \Omega_3$	-8.75e-09	1	0
$\Omega_3 - \Omega_4$	-1.20e-05	1	0
$\Omega_4 - \Omega_5$	-0.0013	0.2040	1
$\Omega_5 - \Omega_6$	-0.0108	0.2673	1

Table 4: Line search for V-cycle 1

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	$\alpha_k$	l.s it
$\Omega_1 - \Omega_2$	-1.95e-08	1	0
$\Omega_2 - \Omega_3$	-2.50e-05	0.2228	1
$\Omega_3 - \Omega_4$	-0.0012	0.0121	3
$\Omega_4 - \Omega_5$	-2.29e-04	0.0135	3
$\Omega_5 - \Omega_6$	-6.99e-05	0.0232	3

Table 5: Line search for V-cycle 4

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	$\alpha_k$	l.s it
$\Omega_1 - \Omega_2$	-1.04e-08	1	0
$\Omega_2 - \Omega_3$	-5.38e-06	0.2758	1
$\Omega_3 - \Omega_4$	-1.41e-04	0.1493	1
$\Omega_4 - \Omega_5$	-1.49e-04	0.0067	4
$\Omega_5 - \Omega_6$	-1.84e-05	0.0722	2

Table 6: Line search for V-cycle 7

Updating	$\nabla J_{\gamma,k}(u_k^{\nu_1})^\top e$	$\alpha_k$	l.s it
$\Omega_1 - \Omega_2$	-4.47e-09	1	0
$\Omega_2 - \Omega_3$	-1.70e-06	1	0
$\Omega_3 - \Omega_4$	-8.52e-05	0.0194	2
$\Omega_4 - \Omega_5$	-2.44e-05	0.0119	3
$\Omega_5 - \Omega_6$	-7.29e-06	0.0619	2

Table 7: Line search for V-cycle 10

## Experiment 2: Comparison between MG/OPT and descent algorithms

In this experiment, we compare the behaviour of the MG/OPT approach versus an optimization algorithm for solving the same problem in the finest grid. In the MG/OPT method these optimization algorithms were used as the underlying optimization algorithms as well. In the following tables we compare the CPU time and the stopping criteria registered for solving the problem described in the previous experiment. Once again, in Tables 8 and 9 we compare one experiment at each row based on the election of the finest grid.

Time (s)		
Mesh	MG/OPT	Gradient method
$\Omega_4$	196.76	-
$\Omega_3$	73.57	2883.55

Table 8: Time comparison, Experiment 1

$ \nabla J_\gamma^\top e $		
Mesh	MG/OPT	Gradient method
$\Omega_4$	3.76e-07	-
$\Omega_3$	2.09e-07	9.92e-07

Table 9: Norm  $|\nabla J_\gamma^\top e|$  comparison, Experiment 1.

As it was expected, it is shown that the MG/OPT performance is more efficient than the descent gradient method. Even if the gradient algorithm is established as the underlying optimization algorithm of the MG/OPT. In what follows we present the same comparison criteria for the preconditioned descent algorithm and the multigrid optimization scheme.

Time (s)		
Mesh	MG/OPT	preconditioned descent algorithm
$\Omega_6$	<b>965.49</b>	2146.37
$\Omega_5$	258.95	71.67
$\Omega_4$	<b>51.23</b>	63.11
$\Omega_3$	10.04	8.30

Table 10: Time comparison, Experiment 1

$ \nabla J_\gamma^\top e $		
Mesh	MG/OPT	preconditioned descent algorithm
$\Omega_6$	7.38e-07	9.35e-07
$\Omega_5$	3.75e-07	3.26e-07
$\Omega_4$	4.70e-07	9.52e-07
$\Omega_3$	3.75e-07	7.84e-07

Table 11: Norm  $|\nabla J_\gamma^\top e|$  comparison, Experiment 1

From Tables 10 and 11 we see that the MG/OPT performs better when reaching the stopping criteria in all cases. The CPU time registered and the tolerance reached are better when working with the MG/OPT algorithm than with the preconditioned descent algorithm at the finest grid  $\Omega_6$ . In this case, CPU time decreases almost in half. However, we can not achieve CPU time savings in all the resolution levels.

### Experiment 3

In this experiment, we set  $p = 1.5$  and  $f = 1$ . We tested the algorithm with two different values of  $g$ . At higher values of  $g$  the plug flow zone is bigger and the execution time of the algorithm increases. Also, as in the previous experiment, the number of iterations is very stable as the mesh resolution is higher. This results are shown in Table 12.

g	Finest mesh	Coarsest mesh	V-cycles	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
0.1	$\Omega_6$	$\Omega_1$	5	8.28e-08	387.50	0.0428
	$\Omega_5$	$\Omega_1$	5	2.46e-08	83.34	0.0427
	$\Omega_4$	$\Omega_1$	3	2.68e-07	11.52	0.0427
	$\Omega_3$	$\Omega_1$	4	5.45e-07	4.02	0.0422
0.4	$\Omega_6$	$\Omega_1$	7	1.29e-07	683.96	9.08e-04
	$\Omega_5$	$\Omega_1$	6	1.63e-07	94.38	9.71e-04
	$\Omega_4$	$\Omega_1$	8	2.57e-08	29.23	8.85e-04
	$\Omega_3$	$\Omega_1$	12	3.02e-07	10.63	7.95e-04

Table 12: Results of the resolution of problem (34) with  $p = 1.5$ ,  $\gamma = 10^3$  and  $f = 1$ .

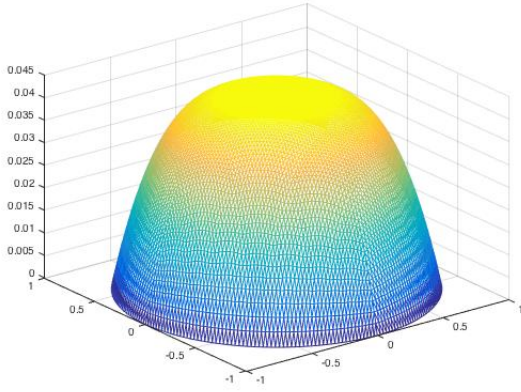


Figure 6: Calculated velocity  $u$  for mesh  $\Omega_6$ . Parameters:  $p = 1.5, g = 0.1, \gamma = 10^3$  and  $\epsilon = 10^{-6}$

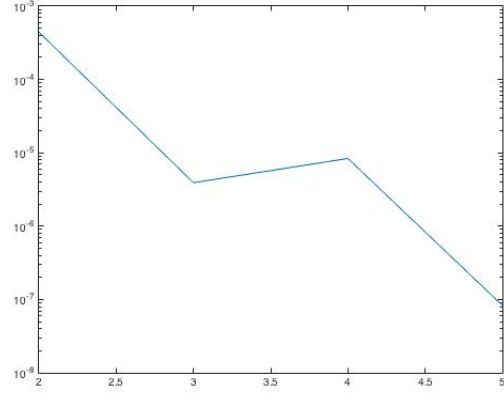


Figure 7: Calculated  $|\nabla J_\gamma e|$  for mesh  $\Omega_6$ .

The resulting velocity function is displayed in Figure 6 for  $g = 0.1$ . Here, we can see that the shear stress transmitted by a fluid layer decreases toward the center of the pipe, which provokes the solid-rigid movement in that area. In Figure 7 the decay of the norm  $|\nabla J_\gamma e|$  is plotted in a logarithmic scale.

#### Experiment 4

For this experiment, we present the behaviour of the algorithm in the unit square domain for the following parameters:  $p = 1.25, g = 0.2, \gamma = 10^3$  and  $f = 3$ . In Figure 8, we can see that in a square domain, the plug flow region is bigger due to the complexity of the domain. In this case, we obtained better results initializing the algorithm with an estimated solution given by one iteration of the full multigrid (FMG) scheme (see [25, Sec. 2.6]) at all refinement levels. We denote by  $r$  the number of MG/OPT cycles inside the FMG approach. All these results are presented in Table 13.

Finest mesh	Coarsest mesh	V-cycles	$r$	$ \nabla J_\gamma e $	Time(s)	Plug flow velocity
$\Omega_5$	$\Omega_1$	1	1	2.07e-07	101.74	0.011
$\Omega_4$	$\Omega_1$	2	1	4.83e-08	26.88	0.011
$\Omega_3$	$\Omega_1$	2	1	7.31e-08	6.88	0.010

Table 13: Results of the resolution of problem (34) with  $p = 1.25, g = 0.2, \gamma = 10^3$  and  $f = 3$ .

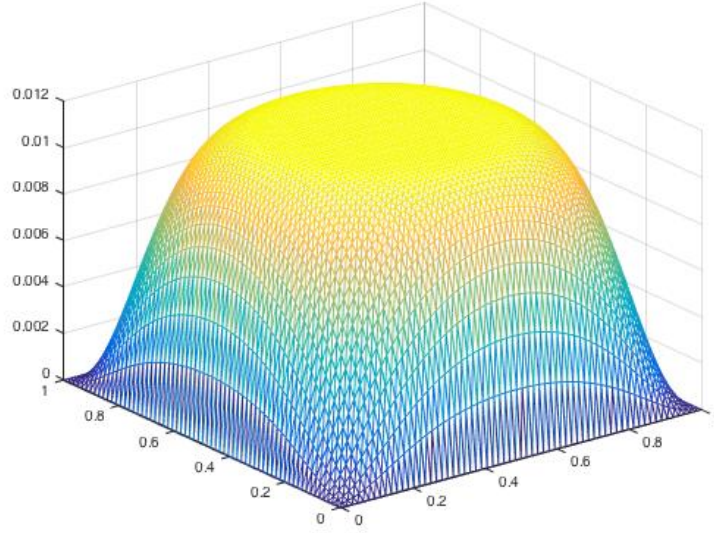


Figure 8: Calculated velocity  $u$  for mesh  $\Omega_5$ . Parameters:  $p = 1.25, g = 0.2, \gamma = 10^3$  and  $f = 3$

#### Experiment 5: Comparison between MG/OPT and descent algorithms

In this experiment, we compare the behaviour of the MG/OPT algorithm versus the preconditioned descent algorithm for solving the same problem described in Experiment 4, in the finest grid.

Mesh	Time (s)	
	MG/OPT	preconditioned descent algorithm
$\Omega_5$	<b>101.74</b>	143.53
$\Omega_4$	<b>26.88</b>	28.08
$\Omega_3$	<b>6.88</b>	8.09

Table 14: Time comparison, Experiment 4

Mesh	$ \nabla J_\gamma^\top e $	
	MG/OPT	preconditioned descent algorithm
$\Omega_5$	2.07e-07	1.38e-07
$\Omega_4$	4.83e-08	5.94e-07
$\Omega_3$	7.31e-08	7.82e-08

Table 15: Norm  $|\nabla J_\gamma^\top e|$  comparison, Experiment 4

Tables 14 and 15 show that we have CPU time savings at all resolution levels. Particularly, we can see a significant reduction of time at the highest resolution level  $\Omega_5$ . The time reduction with respect to the execution time of the preconditioned algorithm is about of the 30%. Regarding the tolerance order achieved at the stopping criteria, both algorithms behaves similarly, with a slight advantage of the MG/OPT method.

## 7.2 Herschel-Bulkley: case $p \geq 2$

### Experiment 6

We now analyze the behaviour of the MG/OPT algorithm in the case  $p > 2$ . The fact that  $p$  increases usually implies instabilities in the performance of the numerical algorithms (see [1, 2, 8, 12]). Also high values of  $g$  are difficult to consider, since the fluid presents more rigid zones along the cross section of the pipe.

Unlike the previous cases, where the MG/OPT algorithm was initialized with the solution of the Poisson problem, in this case we initialize the MG/OPT algorithm with an approximated solution given by the full multigrid (FMG) scheme. In the FMG scheme we apply  $r$  MG/OPT cycles at each level of the algorithm. We set the tolerance in  $10^{-7}$  and the parameters  $p = 4$ ,  $\gamma = 10^3$  and  $f = 3$ .

g	Finest mesh	Coarsest mesh	V-cycles	$r$	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
0.2	$\Omega_5$	$\Omega_1$	8	1	9.22e-07	398.47	0.258
	$\Omega_4$	$\Omega_1$	31	1	3.48e-07	301.23	0.258
	$\Omega_3$	$\Omega_1$	9	2	3.11e-07	21.84	0.258
0.4	$\Omega_5$	$\Omega_1$	175	2	5.59e-07	7826.02	0.153
	$\Omega_4$	$\Omega_1$	51	2	1.80e-07	455.72	0.154
	$\Omega_3$	$\Omega_1$	28	2	8.33e-07	58.47	0.154

Table 16: MG/OPT results of problem (34) with  $p = 4$ ,  $\gamma = 10^3$  and  $f = 3$ .

The last column of Table 16 helps us to test the accuracy of the algorithm when estimating the plug flow velocity with different mesh sizes. The resulting velocity field is displayed in Figure 9. Now we are in the case of a shear-thickening material. Since the shear stress transmitted by a fluid layer decreases toward the center of the pipe, the velocity takes a conical form. The residual presented in Figure 10 shows local convergence in the last iterations.

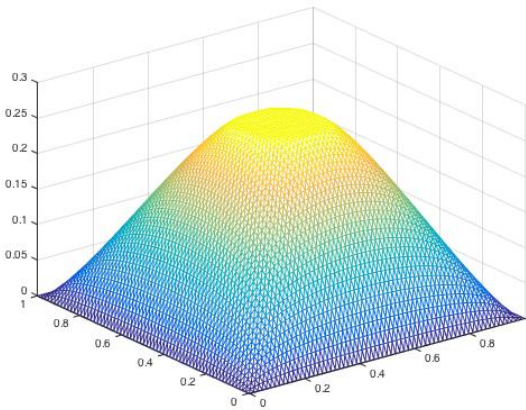


Figure 9: Calculated velocity  $u$  for mesh  $\Omega_5$ . Parameters:  $p = 4$ ,  $g = 0.2$ ,  $\gamma = 10^3$ .

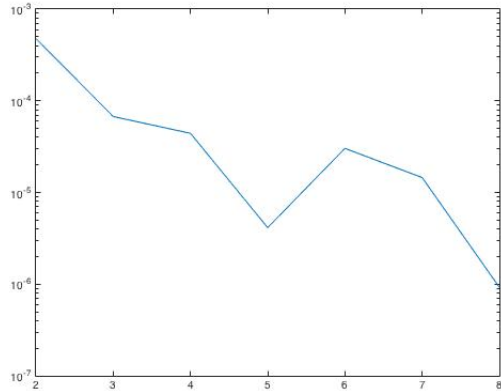


Figure 10: Calculated  $|\nabla J_\gamma^\top e|$  for mesh  $\Omega_5$ .



### 7.3 Bingham

Bingham fluids are viscoplastic materials that can be seen as a particular case of the Herschel-Bulkley model when  $p = 2$ . The main characteristic of Bingham fluids is that when the shear stress exceeds the yield stress, the material presents a linear stress-shear rate relationship (see Figure 3).

#### Experiment 7

The following experiment was carried out in a circular geometry representing the cross section of a pipe, and we consider the following parameters:  $p = 2$ ,  $\gamma = 10^3$  and  $f = 1$ . In Table 17, we can see an example of the algorithm accuracy when determining the plug flow velocity using different mesh sizes. When finer grids are used, there is no difference in the plug flow velocity and we obtain a faster decay of the norm  $|\nabla J_\gamma^\top e|$ . The resulting velocity function is displayed in Figure 11, and we can see the classic behaviour of these materials.

g	Finest mesh	Coarsest mesh	V-cycles	r	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
0.2	$\Omega_5$	$\Omega_1$	94	1	4.18e-07	1723.07	0.090
	$\Omega_4$	$\Omega_1$	13	1	4.96e-07	60.02	0.090
	$\Omega_3$	$\Omega_1$	17	1	6.79e-07	18.64	0.089

Table 17: MG/OPT results for problem (34) with  $p = 2$ ,  $\gamma = 10^3$  and  $f = 1$ .

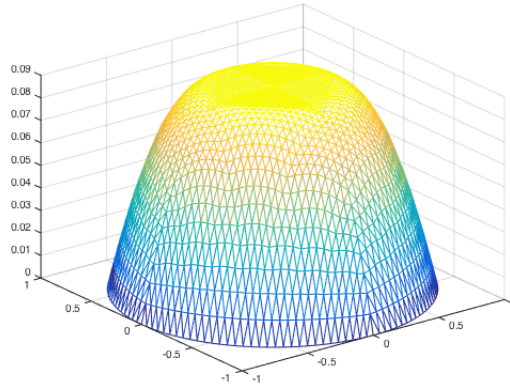


Figure 11: Calculated velocity  $u$  for mesh  $\Omega_5$ . Parameters:  $p = 2, g = 1, \gamma = 10^3$

#### Experiment 8

As  $g$  increases, the plug flow zone increases as well. Thus, we test the MG/OPT algorithm with  $g = 1$  to show this phenomenon in a square cross section. In this experiment we fix the parameters  $\gamma = 10^3$  and  $f = 10$ , and we set the tolerance in  $10^{-5}$ . This experiment was initialized with an estimated solution given by the FMG scheme with  $r$  MG/OPT cycles at all levels of refinement. In Figures 12 and 13 we can notice the difference of the plug flow zone when  $g = 1$  and  $g = 0.2$ , respectively. Also, the execution time when solving this experiment is longer when  $g = 1$  than  $g = 0.2$ . This fact is due to the increasing plug flow and the complexity of the geometry.

$g$	Finest mesh	Coarsest mesh	V-cycles	$r$	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
1	$\Omega_5$	$\Omega_2$	2	1	9.42e-05	152.95	0.2920
	$\Omega_4$	$\Omega_1$	8	1	9.13e-05	85.7	0.2920
	$\Omega_3$	$\Omega_1$	3	2	7.57e-05	9.4	0.2925
0.2	$\Omega_5$	$\Omega_2$	0	1	9.83e-06	56.26	0.6326
	$\Omega_4$	$\Omega_1$	0	1	4.88e-05	10.87	0.6319
	$\Omega_3$	$\Omega_1$	0	1	4.93e-05	9.93	0.6319

Table 18: MG/OPT results of problem (34) with  $p = 2$ ,  $g = 1$ ,  $\gamma = 10^3$  and  $f = 10$ .

An interesting fact in Experiment 8 is exhibited in the fourth column of Table 18 for  $g = 0.2$ . Since  $g$  is small the contribution of the less regular component  $\int_\Omega \psi_\gamma(\nabla u) dx$  of the functional decreases, therefore the plug flow zone is not big with respect to the cross section area. Hence, the minimization problem was uniquely solved by one iteration of the FMG scheme, i.e., the algorithm reached the convergence tolerance in only one iteration.

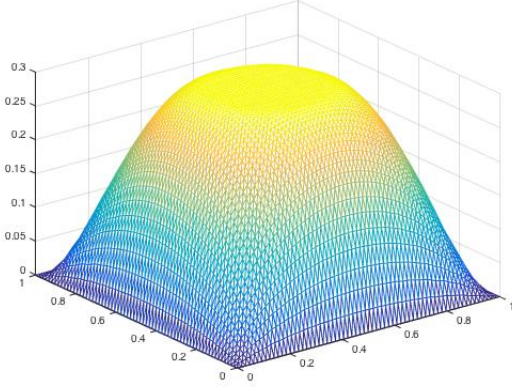


Figure 12: Calculated velocity  $u$  for mesh  $\Omega_5$ . Parameters:  $p = 2$ ,  $g = 1$ ,  $\gamma = 10^3$

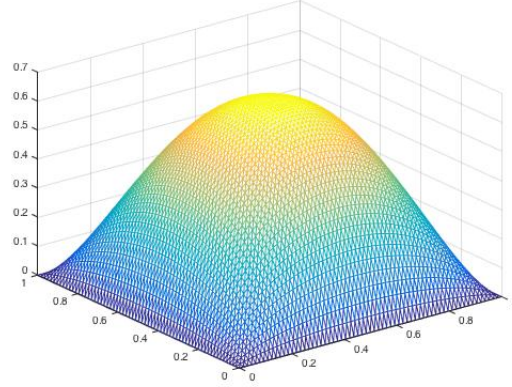


Figure 13: Calculated velocity  $u$  for mesh  $\Omega_5$ . Parameters:  $p = 2$ ,  $g = 0.2$ ,  $\gamma = 10^3$

### Experiment 9: Comparison between MG/OPT and Descent algorithms

In this experiment, we compare the behaviour of the MG/OPT algorithm and the preconditioned descent algorithm when solving the same problem at the finest grid. In Tables 19 and 20, we compare the CPU time and the stopping criteria registered for Experiment 8, with parameters  $p = 2$ ,  $g = 1$ ,  $\gamma = 10^3$  and  $f = 10$ .

Mesh	$ \nabla J_\gamma^\top e $	
	MG/OPT	preconditioned descent algorithm
$\Omega_5$	9.42e-05	5.62e-05
$\Omega_4$	9.13e-05	9.87e-05
$\Omega_3$	7.570e-05	5.61e-05

Table 19: Norm  $|\nabla J_\gamma^\top e|$  comparison.

Time (s)		
Mesh	MG/OPT	preconditioned descent algorithm
$\Omega_5$	<b>152.95</b>	170.79
$\Omega_4$	<b>85.7</b>	103.44
$\Omega_3$	9.4	7.6

Table 20: Time comparison.

From Tables 19 and 20, we can conclude that we can not achieve a decrease of the norm  $|\nabla J_\gamma^\top e|$  with respect to the results obtained by the preconditioned descent algorithm. However, in almost all discretization levels, we have CPU time savings of around the 14% with the MG/OPT algorithm with respect to the execution time of the preconditioned descent algorithm.

## 7.4 Casson

The Casson model is a viscoplastic model that was first developed for modeling printing inks. However, it has also been used to model food flow behaviour such as chocolate and cocoa products [23], and has been applied to biorheology models like hemodynamics and viscometric flows [27].

### Experiment 10

In Tables 21 and 22, we present the results for the Casson fluid flow in a square and circle cross sections, respectively. Both experiments are set with the following parameters:  $\gamma = 10^3$ ,  $f = 1$  and tolerance  $10^{-7}$ . The experiments are initialized with a FMG approximated solution.

g	Finest mesh	Coarsest mesh	V-cycles	$r$	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
0.2	$\Omega_5$	$\Omega_2$	2	2	6.59e-07	321.98	2.68e-04
	$\Omega_5$	$\Omega_1$	4	2	3.23e-07	458.00	2.36e-04
	$\Omega_4$	$\Omega_1$	1	2	7.50e-07	50.03	2.03e-04
	$\Omega_3$	$\Omega_1$	6	2	6.066e-07	28.55	2.06e-04

Table 21: MG/OPT results of problem (34) with  $\gamma = 10^3$  and  $f = 1$ .

g	Finest mesh	Coarsest mesh	V-cycles	$r$	$ \nabla J_\gamma^\top e $	Time(s)	Plug flow velocity
0.2	$\Omega_6$	$\Omega_2$	18	2	9.78e-07	3138.30	0.0150
	$\Omega_6$	$\Omega_1$	30	2	3.25e-08	4553.74	0.0150
	$\Omega_5$	$\Omega_1$	27	2	9.11e-07	829.65	0.0150
	$\Omega_4$	$\Omega_1$	36	2	6.78e-07	243.84	0.0148
	$\Omega_3$	$\Omega_1$	33	2	1.31e-07	56.65	0.0146
0.4	$\Omega_6$	$\Omega_2$	0	2	6.50e-07	377.85	4.91e-04
	$\Omega_6$	$\Omega_1$	1	2	2.60e-07	528.69	4.79e-04
	$\Omega_5$	$\Omega_1$	6	2	5.59e-07	220.84	4.89e-04
	$\Omega_4$	$\Omega_1$	9	2	1.67e-07	59.43	4.85e-04
	$\Omega_3$	$\Omega_1$	6	2	1.034e-07	11.04	4.54e-04

Table 22: MG/OPT results of problem (34) with  $\gamma = 10^3$  and  $f = 1$ .

In these experiments we notice that, if we choose  $\Omega_6$  as the finest mesh and  $\Omega_2$  as the coarsest one instead of  $\Omega_1$ , the algorithm is executed in less time for the same amount of nodes. Moreover, for the parameter  $g = 0.4$ , the algorithm only needs 2 iterations of the FMG scheme to reach the stopping tolerance of  $10^{-7}$ . This behaviour tells us, at least experimentally, that beyond a number of grids used in the multigrid cycles we can not achieve more CPU time savings.

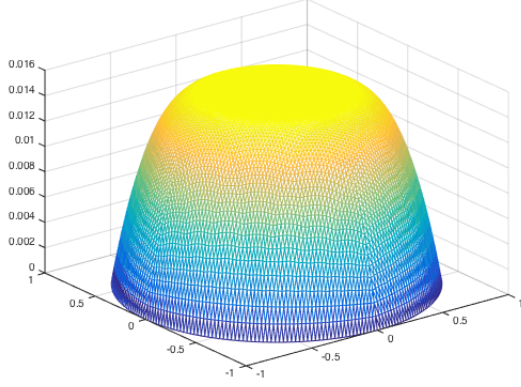


Figure 14: Calculated velocity  $u$  for mesh  $\Omega_6$ . Parameters:  $g = 0.2, \gamma = 10^3$

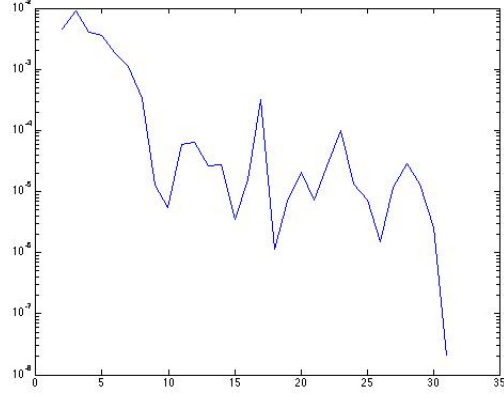


Figure 15: Calculated velocity  $u$  for mesh  $\Omega_6$ . Parameters:  $g = 0.2, \gamma = 10^3$

### Experiment 11: Comparison between MG/OPT and Descent algorithms

In this experiment, we present the results of solving the same problem at the finest grid with the preconditioned descent algorithm and the MG/OPT algorithm. In the following tables we compare the results from Experiment 10, with  $g = 0.4$ , based on the CPU time and the stopping criteria registered.

Mesh	$ \nabla J_\gamma^\top e $	
	MG/OPT	preconditioned descent algorithm
$\Omega_6 - \Omega_2$	6.50e-07	9.8e-07
$\Omega_6 - \Omega_1$	2.60e-07	9.8e-07
$\Omega_5$	5.59 e-07	9.77e-07
$\Omega_4$	1.67e-07	9.14e-07
$\Omega_3$	1.34e-07	9.66e-07

Table 23: Norm  $|\nabla J_\gamma^\top e|$  comparison.

Mesh	Time (s)	
	MG/OPT	preconditioned descent algorithm
$\Omega_6 - \Omega_2$	<b>337.85</b>	2804.14
$\Omega_6 - \Omega_1$	<b>528.69</b>	2804.14
$\Omega_5$	<b>220.84</b>	464.57
$\Omega_4$	<b>59.43</b>	100.51
$\Omega_3$	<b>11.04</b>	19.02

Table 24: Time comparison.

From Table 24, we can conclude that at all discretization levels, we have significant CPU time savings with the MG/OPT algorithm. The average of the time reduction regarding to the execution time of the preconditioned descent algorithm is about the 45% for meshes  $\Omega_5$ ,  $\Omega_4$  and  $\Omega_3$ . Particularly at the finest level  $\Omega_6$  (with a V-cycle with no more than five grids), the time reduction with respect to the time of the preconditioned descent algorithm is about the 87.9%. Also, from Table 23, we notice that the MG/OPT algorithm reaches smaller values for the norm  $|\nabla J_\gamma^\top e|$ , at all the refinement levels, than the preconditioned descent algorithm.

## 8 Conclusions and Outlook

We proposed and analyzed a multigrid for optimization (MG/OPT) algorithm for the numerical solution of a class of quasilinear variational inequalities of the second kind. We analyzed the variational inequality via the minimization of the associated energy functional. First, we regularized the non-differentiable part of the energy by using a Huber regularization approach. Next, we proposed a finite element discretization for the problem, and we deeply analyzed the differentiability of the functional. In particular, we proved that the Jacobian of the discretized functional is slantly differentiable. The MG/OPT algorithm was presented and all of the involved transfer operators analyzed. The convergence of the MG/OPT algorithm was established by using the mean value theorem for slantly differentiable functions and the global convergence of the underlying optimization algorithms. The main issues regarding the implementation of the algorithm were explained, and we described the type of global convergent deepest descent methods used as underlying optimization algorithms. We showed that several classical models for viscoplastic flow correspond to the class of variational inequalities under study. Therefore, we focussed the numerical experiments on this kind of problems. Particularly, we computed the solution for the Herschel-Bulkley, Bingham and Casson models. In all the experiments presented, we appreciated important CPU-time savings, especially when working in the finest meshes. This showed that the MG/OPT algorithm is indeed an efficient tool for dealing with large scale problems.

In order to continue this research, we consider that the study of a more general class of variational inequalities is an interesting perspective. Furthermore, the analysis of more challenging problems, such as the p-Stokes problem, and problems coming from glaciology and geophysics is of great interest. Finally, the analysis and simulation of blood flow models involving the Herschel-Bulkley and Casson structure looks like a very promising field of research.

## References

- [1] J. W. BARRET AND W. B. LIU, *Finite Element Approximation of the p-Laplacian*, Mathematics of Computation, 61 (1993) 523-537.

- [2] R. BERMEJO AND J-. A. INFANTE, *A Multigrid Algorithm for the  $p$ -Laplacian*, SIAM J. Sci. Comput., 21 (2000) 1774-1789.
- [3] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of computation, 21 (1977) 333-390.
- [4] X. CHEN, Z. NASHED AND L. QI, *Smoothing Methods and Semismooth Methods for Nondifferentiable Operator Equations*, SIAM J. Numer. Anal., 38 (2000) 1200-1216.
- [5] R.P. CHHABRA AND J.F. RICHARDSON, *Non-Newtonian Flow and Applied Rheology*, Butterworth-Heinemann, (2008).
- [6] J.E. DENNIS JR AND R.B. SCHNABEL, *Numerical methods for unconstrained optimization and nonlinear equations*, SIAM, 16 (1996)
- [7] G. DUVAUT AND J.L. LIONS, *Inequalities in mechanics and physics*, Springer-Verlag, (1976).
- [8] R. GLOWINSKI AND J-. A. MARROCO, *Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires*, Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique, 9 (1975) 41-76.
- [9] M. S. GOCKENBACH, *Understanding and Implementing the Finite Element Method*, SIAM, U.S.A, 2006.
- [10] S. GONZÁLEZ-ANDRADE, *A Preconditioned Descent Algorithm for Variational Inequalities of the Second Kind Involving the  $p$ -Laplacian Operator*, Computational Optimization and Applications. 66 (2017) 123-162.
- [11] M. HINTERMÜLLER, K. ITO AND K. KUNISCH, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2002) 865-888.
- [12] Y.Q. HUANG, R. LI AND W. LIU, *Preconditioned descent algorithms for  $p$ -Laplacian*, Springer, Journal of Scientific Computing, 32 (2007) 343-371.
- [13] R.R. HUILGOL, *Fluid Mechanics of Viscoplasticity*, Springer Verlag, (2015).
- [14] R.R. HUILGOL AND Z. YOU, *Application of the augmented Lagrangian method to steady pipe flows of Bingham, Casson and Herschel-Bulkley fluids*, Journal of non-newtonian fluid mechanics, 128 (2005) 126-143.
- [15] T. ISAAC, G. STADLER AND O. GHATTAS, *Solution of Nonlinear Stokes Equations Discretized by High-Order Finite Elements on Nonconforming and Anisotropic Meshes, with Applications to Ice Sheet Dynamics*, SIAM J. Sci. Comput., 37 (2015) B804-B833.
- [16] R. KORNUBER, *Monotone multigrid methods for elliptic variational inequalities I*, Numer. Math., 69 (1994) 167-184.
- [17] R. KORNUBER, *Monotone multigrid methods for elliptic variational inequalities II*, Numer. Math., 72 (1996) 481-499.
- [18] R. KORNUBER, *On Constrained Newton Linearization and Multigrid for Variational Inequalities*, Numer. Math., 91 (2002) 699-721.

- [19] O. LASS, M. VALLEJOS, A. BORZÌ AND C. C. DOUGLAS, *Implementation and Analysis of Multigrid Schemes with Finite Elements for Elliptic Optimal Control Problems*, Computing, 84 (2009) 27-48.
- [20] R. M. LEWIS AND S. G. NASH, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM J. Scientific Computing, 26 (2005) 1811-1837.
- [21] S. G. NASH, *A Multigrid Approach to Discretized Optimization Problems*, Optimization Methods and Software, 14 (2000) 99-116.
- [22] J. NOCEDAL AND S. WRIGHT, *Numerical optimization*, Springer Science & Business Media, (2006).
- [23] A. RAO, *Rheology of Fluid, Semisolid, and Solid Foods*, Springer, (2014) 27-61.
- [24] M. RŮŽIČKA, *Electrorheological Fluids: Modeling and Mathematical Theory*, Springer Verlag, (2000).
- [25] U. TROTTEBERG, C. W. OOSTERLEE AND A. SCHULLER, *Multigrid*, Academic press, 2000.
- [26] A. BORZÌ AND M. VALLEJOS, *Multigrid Optimization Methods for Linear and Bilinear Elliptic Optimal Control Problems*, Computing, 82 (2008) 31-52.
- [27] WP. WALAWENDER, TY. CHEN AND DF. CALA, *An approximate Casson fluid model for tube flow of blood*, Biorheology, 12 (1975) 111-119.